

Workshop

Fakta

Kurs:	Systemutvecklare C/C++ Extended 2024
Klass:	SUVx24
Teknikområde:	Programspråket C, in- och utmatning, upprepning

Learning Target

Programmering i C och C++

Grundläggande begrepp inom programmering.

Skapa källkod med C syntax. Tema selektion och boolesk variabel.

Grundläggande in- och utmatning

In- och utmatning av resultat från ett program med hjälp av standardfunktioner i C.

Användning av utvecklingsmiljöer

Använda *Visual Studio Code* (förkortat *VSC*) för att skapa källkod, kompilera och exekvera program skapat med programspråket C.

Introduktion

När vi vill upprepa (**iteration**) kod kan vi göra det med två grundprinciper: *bestämt* eller *obestämt* antal gånger. *Bestämt* antal gånger görs när vi vet hur många gånger vi vill upprepa kod. Vet vi inte på förhand hur många gånger vi ska upprepa ett kodblock kan vi basera det på ett logiskt uttryck. Då använder vi syntaxen för **obestämt** antal gånger med `while` och `do..while`.

Workshop

Förberedelse

Skapa en ny projektmapp (katalog) i den projektmapp du använder för kodning, typ:

`C:\Chas\SUVx24\Workshop7\`

Starta VSC på det sätt du tycker är lättast och öppna den nya projektmappen.

Innehåll

Uppgift 1

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-1.c**

Skapa ett program som

- Skriver ut ditt förnamn och efternamn tio gånger.
- Använd en räknevariabel `n`.

Extra: Gör en inmatning för hur många gånger programmet ska skriva ut namnet.

Uppgift 2

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-2.c**

Använd följande kod:

```
#include <stdio.h>

int main() {

    while(1)
    {
        printf("\nProgramming in C is fun!");
    }

    return 0;
}
```

Kör programmet. Hur många gånger skrivs texten ut? Vad händer?

Avbryt genom att trycka CTRL+C.

Workshop

Uppgift 3a

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-3a.c**

Skapa ett program som

- Skriver ut talen 0 till 100 med texten 'Number =' som en lista.

Uppgift 3b

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-3b.c**

Skapa ett program som

- Skriver ut talen 100 till 0 med texten 'Number =' som en lista.

Uppgift 3c

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-3c.c**

Skapa ett program som

- Först skriver ut alla jämna tal mellan 1 till 99 med texten 'Even number = '.
- Sedan skriver ut alla udda tal mellan 1 till 99 med texten 'Odd number = '.

Workshop

Uppgift 4

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-4.c**

Skapa ett program som

- Skriver ut en multiplikationstabell.
- Frågar användaren vilken tabell som ska skrivas ut och sedan hur många tal från 1 som ska skrivas ut?
- Skriver ut tabellen med rätt antal rader.
- Frågar om användaren vill skriva ut en tabell till?
 - Antingen börjar om eller avslutar programmet.

Extra 1: Försök att få en snygg utskrift av tabellen.

Extra 2: Utöka programmet att även fråga efter hur många multiplikationstabeller att skriva ut. Användare ska kunna ange ett intervall, t ex tabellerna 5 till och med 12.

Uppgift 5

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-5.c**

Skapa ett program som

- Omvandlar mellan temperaturskalorna celsius och fahrenheit.
- Frågar användaren vilken skala att konvertera från samt temperaturvärdet.
- Frågar vilken skala det ska konverteras till.
- Frågar om användaren vill göra en ny konvertering?
 - Antingen börjar om eller avslutar programmet.

Extra: Utöka programmet att även kunna konvertera med temperaturskalan kelvin.

Workshop

Uppgift 6

Att göra:

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-6.c**

Skapa ett program som utökar Workshop 5 - Uppgift 3

- Presenterar menyalternativ på räknesätt (addition, subtraktion, multiplikation, division) samt ett alternativ för att avsluta programmet. Ange en siffra (0 – 4) att mata in respektive val, typ:
 0. Exit program
 1. Add
 2. Subtract
 3. Multiply
 4. Divide
- Frågar efter två (flyt)tal.
- Gör beräkning baserat på val av räknesätt.
 - Om fel inmatning gjordes ska ett felmeddelande om detta skrivas ut.
- Presenterar beräkningen och resultatet.
- Börjar om och presenterar menyn en gång till.
 - Avslutar programmet om alternativ 0 väljs.

Testkör programmet och prova alla alternativen att de fungerar som tänkt.

- Prova även med en felaktig inmatning.
- Vad händer om division med noll utförs?
 - Går det att förebygga?

Workshop

Uppgift 7 - Extra

Gör denna uppgift i mån av tid.

- Skapa och testkör programmet beskrivet nedan med C-kod i en ny fil kallad **workshop7-7.c**

Utöka programmet från **Uppgift 6**.

- Undersök `<math.h>` på webbsidan [C mathematical functions](#). `Math.h` finns i **C Standard Library** och i detta "matte"-bibliotek finns en massa funktioner man kan använda utan att själv behöva skapa dem.
- Försök att integrera funktionerna 'roten ur' och 'upphöjt till' i din kalkylator. När funktionerna används i programmet behövs bara ett inmatat värde.

- *Exempelvis*

```
result = sqrt(inputNumber);
```

- Vad händer om ett negativt tal skrivs in för 'roten ur'? Går det att förebygga?

Extra: Lägg till fler funktioner, t ex absolutbelopp, maximum och minimum, etc.